一种基于元信息的 Android 恶意软件检测方法 *

李江华†，邱　晨

(江西理工大学 信息工程学院, 江西 赣州 341000)

摘　要：Android 应用普遍具有比所属类型更多的功能，需要获取更多的权限，过多的权限可能带来一定的安全隐患。针对这类问题，提出一种基于元信息的 Android 恶意软件检测方法。首先，通过对 Android 应用程序描述进行 LDA 主题提取，实现数据降维，使用 K-means 聚类算法按照功能类型对应用程序分组；然后，对属于同一功能类型的所有应用程序，提取其权限信息，以权限特征为研究对象，使用 kNN 算法进行 Android 恶意软件的分类检测。实验结果获得 94.81%的平均准确率，证明方法的有效性和高准确率。

关键词：Android 恶意软件检测；元信息；应用程序描述；权限特征

中图分类号：TP311　　doi: 10.3969/j.issn.1001-3695.2018.04.0312

Android malware detection method based on meta-information

Li Jianghua†, Qiu Chen

(*School of Information Engineering Jiangxi University of Science & Technology*, *Ganzhou Jiangxi* 341000, *China*)

Abstract: Many applications have more functions than their types, and they need to acquire more permissions. Excessive permissions may bring some security risks. To address these issues, this paper proposes an Android malware detection method based on meta information. First, through the description of Android application of LDA theme extraction, the implementation of data dimensionality reduction, using the k-means clustering algorithm in accordance with the functional type of the application group; Then, for all applications belonging to the same functional type, extract their permission information, and take the permission features as the research object, using KNN algorithm to classify and detect the malicious software of Android. The experimental results obtained the average accuracy of 94.81% and proved the validity and high accuracy of the method.

Key words: Android malware detection; meta information; application description; permission features

0　引言

Android 应用程序元信息，是指用户在下载和安装应用程序之前看到的信息[1]，如应用程序描述、评分等。Android 应用程序元信息包含的信息种类非常多；同时，由于各大应用市场的管理方式、经营方式和设计方式的不同，用户看到的信息也有所区别。

Android 应用程序描述的作用是对应用程序进行介绍，说明其功能。需要注意的是，很多应用程序提供了比所属类型更多的功能，这在应用程序描述中才能够看到。

应用程序要执行其功能，需要相应的权限支持，很多应用具有比所属类型更多的功能，需要获取更多的权限，过多的权限可能带来一定的安全隐患。要确定这样的应用是否为恶意应用，需要进一步的分析检测。

1　相关研究

当前，在进行 Android 恶意软件检测方法的研究时，多数研究聚焦于应用程序的权限[2,3]、Java 代码[4~6]等静态特征和系统调用[7]、网络流量[8,9]等动态特征上，针对动态特征进行研究或者使用动态特征和静态特征相结合的方式[10]赢得了更多科研人员的青睐。

周裕娟等人[11]提出了一种使用权限信息和权限提升威胁信息作为特征进行 Android 恶意软件检测的方法。Enck 等人[12]提出了一种 Kirin 安全规则进行 Android 恶意软件检测。Felt 等人[13]认为应用程序的任何行为都是与 API 的调用有关的，而 API 调用又可以映射到权限，他们在研究中使用了一种可以自动化检测 Android API 调用的工具，以便构建检测超额权限所必需的权限映射，用实验证明有三分之一的应用程序存在过度申请权限的情况。杨欢等人[14]利用关联规则挖掘的方式试图找

收稿日期：2018-04-08；**修回日期：**2018-05-29　**基金项目：**国家自然科学基金资助项目（61463021，61762046）；江西省教育厅科技项目（GJJ160599，GJJ170516）

作者简介：李江华（1976-），男（通信作者），河南新野人，博士，主要研究方向为信息安全、语义 Web、大数据分析与处理（4912170@qq.com）；邱晨（1992-），男，江西九江人，硕士，主要研究方向为信息安全、机器学习．

到权限的组合关系与恶意应用程序相关的规则，基于这个规则进行恶意软件检测，最后获得了 87%的准确率。文伟平等人[10]在其研究中也给出了一套危险权限组合规则，如权限组合 RECEIVE_BOOT_COMPLETED + INTERNET + ACCESS_COARSE_LOCATION，会泄露用户的位置，但是某名为“Puppet Football Fighters-Steampunk Soccer”体育游戏类应用，其权限列表（图 1）就包含这种组合，使用权威检测网站 VirusTotal 对其进行检测，检测结果为良性软件。这说明危险组合权限规则并不具有普遍适用性。

权限
android.permission.ACCESS_COARSE_LOCATION
android.permission.ACCESS_NETWORK_STATE
android.permission.ACCESS_WIFI_STATE
android.permission.INTERNET
android.permission.RECEIVE_BOOT_COMPLETED
android.permission.VIBRATE
android.permission.WAKE_LOCK
android.permission.WRITE_EXTERNAL_STORAGE
com.android.vending.BILLING
com.google.android.c2dm.permission.RECEIVE
com.noxplay.PuppetFootballFighters.permission.C2D_MESSAGE

图 1　某应用程序权限列表

受限于硬件和软件技术，一些方法即使取得了较好的效果，也不便于推广和应用。因此，基于 Android 应用程序元信息的恶意软件检测受到了一部分研究者的关注，开展了一定的研究。Teufl 等人[15]使用复杂的知识发现过程和精益统计的方法来分析从 Google 官方应用市场 Google Play 收集的元信息，并使用机器学习算法进行分类。Martin 等人[16]在其研究中探索了一种名为 ADROIT 的方法，使用了多种元信息作为实验数据，声称最终获得了 93.67%的精度。这类研究中采用的元信息种类众多，处理难度较高。基于此，Gorla 等人[17]提出了一种采用元信息种类较少的方法，通过应用程序描述确定应用程序的主题，主题词由研究者自行决定，通过对应用程序的主题进行聚类，然后确定与主题相关的 API 调用异常，他们在 MalGenome[18]的数据样本上正确识别了 56%的恶意应用程序。

在使用较少元信息种类的前提下，为了提高检测率，也为确定具有更多功能的应用是否为恶意应用，本文提出一种新的基于元信息的 Android 恶意软件检测方法，以 Android 应用程序元信息为研究对象，进行 Android 恶意软件检测方法的研究。

2　本文方法

本文方法选择的研究对象是 Android 应用程序描述和权限，它们都属于 Android 应用程序元信息。方法有两个主要步骤：

a）根据 Android 应用程序描述确定应用程序的功能类型，按照功能对应用程序分组。

Android 应用程序描述用于介绍应用程序的功能，从中提取出描述功能的关键词，然后使用聚类算法进行聚类，可确定应用程序的功能类型，进而达到按照功能对应用程序分组的目的。举例说明，设有 A、B、C、D 四个应用程序，它们的功能如表 1 所示，经过聚类处理后将 A、B、C、D 四个应用程序按照功能进行分组，结果如表 2 所示，其中每个组的名称用功能表示。

表 1　功能列表

应用程序	功能			
A	功能 1	功能 2	功能 3	功能 4
B	功能 1	功能 5	功能 6	—
C	功能 2	功能 6	—	—
D	功能 4	功能 6	功能 7	—

表 2　聚类后分组结果

功能	应用程序	功能	应用程序
功能 1	A，B	功能 5	B
功能 2	A，C	功能 6	B，C，D
功能 3	A	功能 7	D
功能 4	A，D		

b）对于同一功能类型的所有应用程序，提取其权限信息，以权限特征为研究对象，进行 Android 恶意软件的分类检测。

权限是应用程序获得某种功能的能力的标志，因此，相同功能类型的应用程序应当具有相同或相似的权限列表，若某个应用程序具有多余的权限，那么该应用程序就有可能是恶意性的，通过对权限进行分析，以确定其是否为恶意应用。分析过程如下：

a)针对同一功能类型的所有应用程序，统计出其权限列表中不同权限的名称（$p_1, p_2, \cdots, p_m$） 以及个数 m 。

b)给权限的名称随机确定一个排列顺序或者按照既定的顺序（如字典顺序）排序。

c)对于某个应用程序，将其权限列表与排好序的权限列表依次进行对照，匹配到则记“1”，其余记“0” “1”和“0”的数目之和等于之前统计到的不同权限的个数 m，由此得到了该应用程序关于权限信息的 m 维向量。重复此步骤，即可获得所有应用程序的权限特征向量。

d)同组的所有应用程序构成样本集，对应用程序样本集进行划分，得到训练样本集和测试样本集。

e)将权限信息向量化后，计算训练样本集中的权限向量与测试样本集中某待测样本权限向量之间的相似度。若训练样本集中某样本权限向量与待测样本权限向量的相似度达到某个预设的度量标准，则认为待测样本与该样本的性质相同。若该样本是恶意应用，则待测样本是恶意应用；若该样本是良性应用，则待测样本是良性应用。

举例说明，如表 2 所示，功能 6 代表的组别中有 B、C、D 三个应用程序，要对功能 6 这一组的应用程序进行分析，那么这三个应用程序就构成了应用程序样本集，将这个应用程序样本集划分成训练样本集和测试样本集，应用程序 B、C 构成训练样本集，应用程序 D 构成测试样本集。假设 B 为恶意应用，C 为良性应用。对权限信息向量化后，若 B 的权限向量是

$(x_{11}, x_{12}, \cdots, x_{1n})$，C 的权限向量是 $(x_{21}, x_{22}, \cdots, x_{2n})$，D 的权限向量是 $(x_{31}, x_{32}, \cdots, x_{3n})$。针对待测样本 D，计算其与训练样本集中每个样本即 B、C 的相似度 S_{BD}、S_{CD}。若 S_{BD} 达到预设度量标准 η，则 D 的性质与 B 相同，D 为恶意应用；若 S_{CD} 达到预设度量标准 η，则 D 的性质与 C 相同，D 为良性应用。

3　实验及分析

3.1　实验方案

针对上述方法进行实验验证。本文实验的运行环境如下：

a）硬件环境。CORE i3 处理器、Windows7 64 位操作系统、内存为 4 GB 的笔记本电脑。

b）软件环境。编程环境为 PyCharm 5.0.3，编程语言为 Python，编译器为 Python 3.5。

获取到实验数据后，经过数据预处理，将分组实验使用的 Android 应用程序描述进行文本清洗，处理过后即可得到关于 Android 应用程序描述的语料库，使用 LDA 主题模型进行主题抽取，然后将主题词向量化。将应用程序描述转换为 K-means 算法能够处理的数据后，使用 K-means 算法进行聚类处理，得到具有相似性质的 k 个簇群，通过 LDA 主题模型处理的结果可得到这些簇群中的主题词及其所对应的应用程序。

按照功能类型分组应用程序后，采用权限特征，进行分类检测实验。所有实验数据样本都先通过权威网站 VirusTotal 确定应用程序样本的性质（良性或恶意），然后提取应用程序样本的权限信息，以权限特征为研究对象，使用 kNN 算法进行分类检测，将检测结果与由 VirusTotal 得到的结果作对比，以验证分类算法的检测效果。

3.2　实验数据

本文所有数据均来自于 Aptoide 应用市场，通过网络爬虫获得，Aptoide 应用市场提供的 Android 应用程序元信息条目比较全。所获得的信息包括应用程序描述、评分、兼容性（Android 版本）、权限、应用程序名称、大小、应用程序版本号、发布时间、适用最小屏幕、支持的处理器型号、程序包 ID、MD5 值、签名 SHA1、开发人员代号、组织、地点、国家、省市等信息，随机爬取了共计 17 812 条 app 元信息记录。

Aptoide 应用市场是开放性市场，它面向全世界开放，所有人都可以在其中发布自己开发的应用程序，因此，这个应用市场包含世界各国的开发人员的作品，应用程序描述也是由各种语言表述。同时，很多应用程序缺少部分信息，由于不清楚这种情况对实验结果是否有影响，所以有必要对这种情况进行处理。

为避免由于信息的缺失对实验结果产生不可预知的影响，删除了部分信息缺失的记录；同时删除了没有应用程序描述或应用程序描述用非中文和英文表达的记录，仅保留了由中文或英文表达 Android 应用程序描述的记录。

经过上述处理过后，原本的 17 812 条记录剩余 5 000 多条记录，其中包含应用程序描述、权限和应用程序名称，去除权限列表中只包含 Normal 类权限的应用程序，因为它们都是良性应用。这就是本文实验最终使用的样本数据。

3.3　数据预处理

应用程序描述是文本数据，需要经过处理才能被计算机识别。

经过清洗过程后，得到关于应用程序描述的语料库，然后针对语料库使用 LDA 主题模型进行主题抽取。经过 LDA 主题模型处理后能够得到文档与主题词的关系（亲和度），这种关系用概率（百分比）表示，如表 3 所示。

表 3　文档、主题词关系

	主题词 1	主题词 2	主题词 3	主题词 4
文档 1	0.6	0.3	0.1	0
文档 2	0	0.5	0.3	0.2
文档 3	0.2	0	0	0.8

通过 LDA 主题模型不仅获得了文档与主题词的关系，同时也对数据进行了降维处理。到这一步，依然没有将样本处理成为 K-means 算法能够处理的数据，接下来对主题词进行向量化，将主题词转换为特征向量。

举例说明，现在有三个句子，要将其向量化：

a）Apple is delicious.

b）I love apple.

c）Apple is delicious and I love apple.

通过删除特殊字符，去除停用词等清洗过程后，这三句话就转换为如表 4 所示的表格。第一句话向量化表示为 $[1,0,1,0]$，第二句表示为 $[1,1,0,1]$，第三句表示为 $[1,1,1,1]$，至此，三个句子向量化完成。

表 4　主题词向量化

	apple	I	delicious	love
句 1	1	0	1	0
句 2	1	1	0	1
句 3	1	1	1	1

事实上，针对清洗过后得到的语料库直接进行向量化也可以达到目的。但是本文并没有这样做，而是选择先进行 LDA 主题抽取，然后再向量化主题词。原因在于聚类算法在涉及少量特征时效果更好。因此，将描述转换为主题对于获得更好的聚类结果至关重要。

经过 LDA 主题模型进行主题提取后可以确定文档和主题关系、主题与词关系，同时可以输出提取到的主题词。

3.4　聚类分析

通常一个应用程序最多具有 5 种不同类型的功能，故设定每篇文档最多与 5 个主题有关，但不排除这 5 个主题词为同一个的可能性，这说明该应用程序没有多余的功能。经过 LDA 主题模型处理后，可得到应用程序描述文档与主题之间的亲和度关系向量，这个向量作为 K-means 聚类算法的输入。实验中设置聚类质心数 k，取值范围为 2~30。结果表明，当 k 值为 14 时得到较好的聚类结果。根据聚类结果将应用程序分组，结

果如表 5 所示。

表 5　应用程序分组结果

类型	数量	类型	数量
connection	739	account and payment	192
social	574	browser	156
language	230	news and sharing	961
manager	1269	player	753
wallpaper	351	navigation	198
game	2836	theme and wallpaper	142
shopping	183	advertisement	398

“game”类应用其实包含“challenge game”“sport game”“battle game”“car game”等多种，这些不同的“game”类应用只有玩法或操作上的区别，经过聚类算法后，它们被划分为同一个类型。表 5 中，“theme and wallpaper”没有被划分到“wallpaper”类中。与“game”类应用不同，当改变手机的壁纸时，只会让手机屏幕桌面的背景发生改变，但是改变主题时，会引起手机里的图标、桌面背景、整体颜色发生改变，这是具有很大区别的两种应用。

分析这些应用，在国内，“connection”类最常用的应该是“wifi 万能钥匙”，“social”类如“微信”“QQ”等，“shopping”类如“淘宝”“京东”等，它们都是与日常生活紧密相关的应用类型，这些应用拥有庞大的用户群体，因而更容易成为恶意应用开发者的目标。

3.5　判别检测

在进行判别检测实验前首先通过 VirusTotal 网站确定了应用程序的性质：恶意性或良性，5 000 多个应用程序中有 624 个恶意应用，其在各个类别中的分布如表 6 所示。

表 6　各类型恶意应用分布

功能类型	数量	恶意应用数量
connection	739	18
social	574	73
language	230	0
manager	1269	352
wallpaper	351	0
game	2836	23
shopping	183	17
account and payment	192	36
browser	156	1
news and sharing	961	135
player	753	264
navigation	198	61
theme and wallpaper	142	13
advertisement	398	109
总和	8982	1102

由表 6 可知，恶意软件主要分布在“manager”“player”“news and sharing”“advertisement”类中，其余类型中虽然也有恶意软件，但是数量较少，不足以作为机器学习算法的训练样本。表 6 中第二、三列数据的总和均大于实际样本数量和实际恶意软件数量，导致这种结果的原因是部分应用程序不只有一个功能，被划分到多个类型中；同时如果应用程序为恶意应用，由于类型的多样性，可能会被重复计算多次。

3.5.1 样本不均衡问题处理方案

如果样本数量不均衡，对结果的影响巨大。比如训练样本集中包含两类数据样本，但是其中一类样本的数量比另一类样本数量多，按照 kNN 算法的多数投票原则，因为 k 个邻近点中正确类标签的数量比错误类标签的数量少而可能导致分类错误。

针对因为样本数量不均衡导致的误分类的问题，使用对训练样本数据进行处理的方式，比较常用的方法是欠抽样方法和过抽样方法。欠抽样方法就是为了让样本数据多的一方参照数据少的一方，舍弃部分样本数据；而过抽样方法则是对数据样本采用重复抽样的方式，以达到让样本数据少的一方向数据多的一方并齐的目的。

在集成方法中有一种 bootstrap 抽样方法，如果原始数据集中有 100 个数据，现在要创建用于训练的新数据集，这个新数据集是通过在原始数据集中有放回的随机抽样 100 次得到的。

将欠抽样方式、过抽样方式与 bootstrap 抽样方式相结合，随机从样本数据多的一方选取数据样本组成新的训练样本数据。新的训练样本数据集中样本的数量接近或等于样本数据少的一方的样本数量，从而使样本数据数量均衡，进而避免因为样本数量不均衡导致的误分类问题。其次，欠抽样方式可能导致某些具有较大价值的样本未被选入到新数据样本里面，为了避免这个问题的发生，可以采取多次抽样进行实验的方式达到目的。设样本数据多的一方数据集为 A ，样本数据少的一方数据集为 B ，若 A 中样本数量是 B 中样本数量的 t 倍，抽样方法如下：

a）若 t 为正整数，则可以采用在 A 中随机不放回抽样的方式，将 A 拆分成 t 个新数据集，每个新数据集分别与 B 组合得到用于 kNN 算法的 t 个训练数据集，这意味着选定 k 值后，kNN 算法需要运行 t 次，得到同一个数据的 t 个测试结果，对这 t 个结果采用多数投票的方式决定最后分类结果；

b）若 t 不是正整数，则可以将 A 拆分成 $\lceil t \rceil$ 个新数据集，前 $\lceil t \rceil - 1$ 个数据集的生成方式与 a）中相同，第 $\lceil t \rceil$ 个数据集中不够的数据样本可以在原始数据集中随机不放回地选取补充。每个新数据集分别与 B 组合得到用于 kNN 算法的 $\lceil t \rceil$ 个训练数据集，这意味着选定 k 值后，kNN 算法需要运行 $\lceil t \rceil$ 次，得到同一个数据的 $\lceil t \rceil$ 个测试结果，对这 $\lceil t \rceil$ 个结果采用多数投票的方式决定最后分类结果。

3.5.2 结果评估及分析

针对“manager”“player”“news and sharing”“advertisement”4 类应用程序样本，都采用相同的样本处理方式，将良性应用样本和恶意应用样本都以 2:1 的比例划分成训练数据样本和测试数据样本，则上述 4 类应用程序的划分结果如表 7 所示。

表 7　各类样本划分结果

类型	良性应用		恶意应用	
	训练样本数	测试样本数	训练样本数	测试样本数
manager	846	423	235	117
player	641	320	90	45
news and sharing	502	251	176	88
advertisement	265	133	73	36

分析表 7 中的数据，发现良性应用样本数量比恶意应用样本数量多，“manager”类中良性应用样本数量是恶意应用样本数量的 3.6 倍，“player”类中良性应用样本数量是恶意应用样本数量的 7.12 倍，“news and sharing”类中良性应用样本数量是恶意应用样本数量的 2.85 倍，“advertisement”类中良性应用样本数量是恶意应用样本数量的 3.6 倍，样本数量极为不均衡。为了避免因为样本数量差距太大造成分类结果错误率提升的问题，可以采用 3.5.1 节所述方案重新创建多个训练样本数据集。则“manager”“player”“news and sharing”和“advertisement”类的训练样本集数量分别为 4、8、3、4 个，测试样本保持不变。重新划分训练集的结果如表 8~11 所示。

表 8　manager 类样本划分结果

应用程序性质	良性应用			恶意应用	
样本集	训练样本集1	训练样本集2	测试样本集	训练样本集	测试样本集
数量	235	235	423	235	117
样本集	训练样本集3	训练样本集4	测试样本集	训练样本集	测试样本集
数量	235	235	423	235	117

表 9　player 类样本划分结果

应用程序性质	良性应用			恶意应用	
样本集	训练样本集1	训练样本集2	测试样本集	训练样本集	测试样本集
数量	90	90	320	90	45
样本集	训练样本集3	训练样本集4	测试样本集	训练样本集	测试样本集
数量	90	90	320	90	45
样本集	训练样本集5	训练样本集6	测试样本集	训练样本集	测试样本集
数量	90	90	320	90	45
样本集	训练样本集7	训练样本集8	测试样本集	训练样本集	测试样本集
数量	90	90	320	90	45

表 10　news and sharing 类样本划分结果

应用程序性质	良性应用				恶意应用	
样本集	训练样本集1	训练样本集2	训练样本集3	测试样本集	训练样本集	测试样本集
数量	176	176	176	251	176	88

表 11　advertisement 类样本划分结果

应用程序性质	良性应用			恶意应用	
样本集	训练样本集1	训练样本集2	测试样本集	训练样本集	测试样本集
数量	73	73	133	73	36
样本集	训练样本集3	训练样本集4	测试样本集	训练样本集	测试样本集
数量	73	73	133	73	36

将每个类型的多个训练样本的权限特征向量化后输入到 kNN 算法，再使用测试集进行分类测试。实验过程中，为避免出现归属为两个类别的邻近点数量相等的情况，k 值均在奇数中选取，对四种类型的应用程序进行分类时 k 的取值范围统一为 $\{1,3,5\}$ 。

为评估分类结果，假设 TP 代表正确分类良性应用的数目，FP 代表恶意应用被错误分类为良性应用的数目，TN 代表正确分类恶意应用的数目，FN 代表良性应用被错误分类为恶意应用的数目，则按照分类实验评估方法，kNN 算法对每个功能类型的分类结果如表 12 和 13 所示。

表 12　manager 类和 player 类分类结果

k 值	manager类					player类				
	TP	FP	TN	FN	准确率ACC	TP	FP	TN	FN	准确率ACC
1	403	13	104	20	0.9388	310	7	38	10	0.9534
3	408	8	109	15	0.9574	307	10	35	13	0.9370
5	405	10	107	18	0.9481	303	11	34	17	0.9233

表 13　news and sharing 类和 advertisement 类分类结果

k 值	news and sharing类					advertisement类				
	TP	FP	TN	FN	准确率ACC	TP	FP	TN	FN	准确率ACC
1	240	10	78	11	0.9381	125	8	28	8	0.9053
3	242	8	80	9	0.9499	128	6	30	5	0.9349
5	243	8	80	8	0.9528	126	9	27	7	0.9053

对上述实验结果进行分析可知，四类应用最高准确率对应的 k 值并不相同，“manager”类在 k 值为 3 时准确率最高，为 95.74%；“player”类获得最高检测率为 95.34%，此时 k 值为 1；“news and sharing” 类获得最高准确率为 95.28%，此时 k 值为 5；“advertisement” 类获得最高准确率为 93.49%，此时 k 值为 3。四类应用的平均准确率分别为 94.81%、93.79%、94.69%、91.51%。

表 14　相关方法对比

检测方法	检测率
文献[11]方法	92.86%
文献[14]方法	87%
文献[16]方法	93.67%
文献[17]方法	81.18%
本文方法	94.81%

表 14 展示了本文方法与近几年相关工作的对比情况。文献[11,14]的方法都使用权限特征，不同的是文献[11]使用权限信息和权限提升威胁信息作为研究对象，文献[14]试图挖掘权限组合关系与恶意应用相关的规则。这两种方法均具有一定的检测效果，缺点是必须先反编译应用程序，然后才能提取到所需的权限信息。

文献[16,17]的方法均涉及元信息的使用，元信息可以直接从应用市场提取，不需要经过如反编译等过程，更加方便。文献[17]的方法中，检测模型的建立是基于大量良性应用的，缺少恶意应用样本，导致检测结果存在较多的误判。与同样使用元信息进行恶意应用检测，且数据来源也是 Aptoide 应用市场的文献[16]的 ADROIT[16]方法相比，本文检测率比较接近，且有所提高，同时比单独采用权限信息作为研究对象的方法，检测准确率明显提高。证明提出的方法是有效的，并且获得了较高的准确率。

4　结束语

Android 恶意软件的检测是一个重要的研究方向，当前的研究主要集中在基于静态特征和动态特征的方法研究上。本文从应用程序描述元信息的角度，提出了一种 Android 恶意软件检测方法。经过实验验证，本文提出的方法是有效的。本文提出的检测方法使用了经典的聚类和分类算法，算法检测的准确性，一定程度上依赖于聚类和分类的效果；算法检测的准确性，仍有一定的改进空间，这将是本文下一步的工作。

参考文献：

[1] Feizollah A, Anuar N B, Salleh R, *et al.* A review on feature selection in mobile malware detection [J]. Digital Investigation the International Journal of Digital Forensics & Incident Response, 2015, 13 (C): 22-37.

[2] Sarma B P, Li N, Gates C, *et al.* Android permissions: a perspective combining risks and benefits [C]// Proc of ACM Symposium on Access Control Models and Technologies. New York: ACM Press, 2012: 13-22.

[3] 杨宏宇, 徐晋. Android 恶意软件静态检测模型 [J]. 吉林大学学报: 工学版, 2018, 48 (2): 564-570. (Yang Hongyu, Xu Jin. An Android malware static detection model [J]. Journal of Jilin University: Engineering and Technology Edition, 2018, 48 (2): 564-570.)

[4] Chen Jian, Alalfi M H, Dean T R, *et al.* Detecting Android malware using clone detection [J]. 计算机科学技术学报: 英文版, 2015, 30 (5): 942-956.

[5] 李根. Android 系统恶意代码检测技术研究 [D]. 哈尔滨: 哈尔滨工业大学, 2014. (Li Gen. Research on malware detection technology for Android [D]. Harbin: Harbin Institute of Technology, 2014.)

[6] 韩金, 单征, 赵炳麟, 等. 基于软件基因的 Android 恶意软件检测与分类 [J/OL]. 计算机应用研究, 2019, 36 (6): 1-9 [2018-03-16]. http://www. arocmag. com/article/02-2019-06-039. html. (Han Jing, Shan Zheng, Zhao Binglin, *et al.* Detection and classification of Android malware based on malware gene [J]. Application Research of Computers, 2019, 36 (6): 1-9 [2018-03-16]. http://www. arocmag. com/article/02-2019-06-039. html.)

[7] 蔡志标, 彭新光. 基于系统调用的 Android 恶意软件检测 [J]. 计算机工程与设计, 2013, 34 (11): 3757-3761. (Cai Zhibiao, Peng Xinguang. Detection of Android malware based on system calls [J]. Computer Engineering and Design, 2013, 34 (11): 3757-3761.)

[8] Malik J, Kaushal R. CREDROID: Android malware detection by network traffic analysis [C]// Proc of ACM Workshop on Privacy-Aware Mobile Computing. New York: ACM Press, 2016: 28-36.

[9] 吴非，裴源，吴向前．一种改进贝叶斯模型的 Android 恶意软件流量特征分析技术 [J]. 小型微型计算机系统, 2018, 39 (2): 230-234. (Wu Fei, Pei Yuan, Wu Xiangqian. Android malware traffic feature analysis technique based on improved Bayesian model [J]. Journal of Chinese Computer Systems, 2018, 39 (2): 230-234.)

[10] 文伟平，梅瑞，宁戈，等．Android 恶意软件检测技术分析和应用研究 [J]. 通信学报, 2014, 35 (8): 78-85, 94. (Wen Weiping, Mei Rui, Ning Ge, *et al.* Malware detection technology analysis and applied research of android platform [J]. Journal on Communications, 2014, 35 (8): 78-85, 94.)

[11] 周裕娟，张红梅，张向利，等．基于 Android 权限信息的恶意软件检测 [J]. 计算机应用研究, 2015, 32 (10): 3036-3040. (Zhou Yujuan, Zhang Hongmei, Zhang Xianli, *et al.* Malware detection based on Android permission information [J]. Application Research of Computers, 2015, 32 (10): 3036-3040.)

[12] Enck W, Ongtang M, McDaniel P, *et al.* On lightweight mobile phone application certification [C]// Proc of the 16th ACM Conference on Computer and Communications Security. New York: ACM Press, 2009: 235-245.

[13] Felt A P, Chin E, Hanna S, *et al.* Android permissions demystified [C]// Proc of ACM Conference on Computer and Communications Security. New York: ACM Press, 2011: 627-638.

[14] 杨欢，张玉清，胡予濮，等．基于权限频繁模式挖掘算法的 Android 恶意应用检测方法 [J]. 通信学报, 2013, 34 (S1): 106-115. (Yang Huan, Zhang Yuqing, Hu Yupu, *et al.* Android malware detection method based on permission sequential pattern mining algorithm [J]. Journal on Communications, 2013, 34 (S1): 106-115.)

[15] Teufl P, Ferk M, Fitzek A, *et al.* Malware detection by applying knowledge discovery processes to application metadata on the Android market (Google play) [J]. Security & Communication Networks, 2016, 9 (5): 389-419.

[16] Martín A, Calleja A, Menéndez H D, *et al.* ADROIT: Android malware detection using meta-information [C]// Proc of IEEE Symposium Series on Computational Intelligence. Piscataway, NJ: IEEE Press, 2017: 1-8.

[17] Gorla A, Tavecchia I, Gross F, *et al.* Checking App behavior against App descriptions [C]// Proc of International Conference on Software Engineering. New York: ACM Press, 2014: 1025-1035.

[18] Jiang X, Zhou Y. Dissecting Android malware: characterization and evolution [C]// Proc of IEEE Symposium on Security and Privacy. Washington DC: IEEE Press, 2012: 95-109.